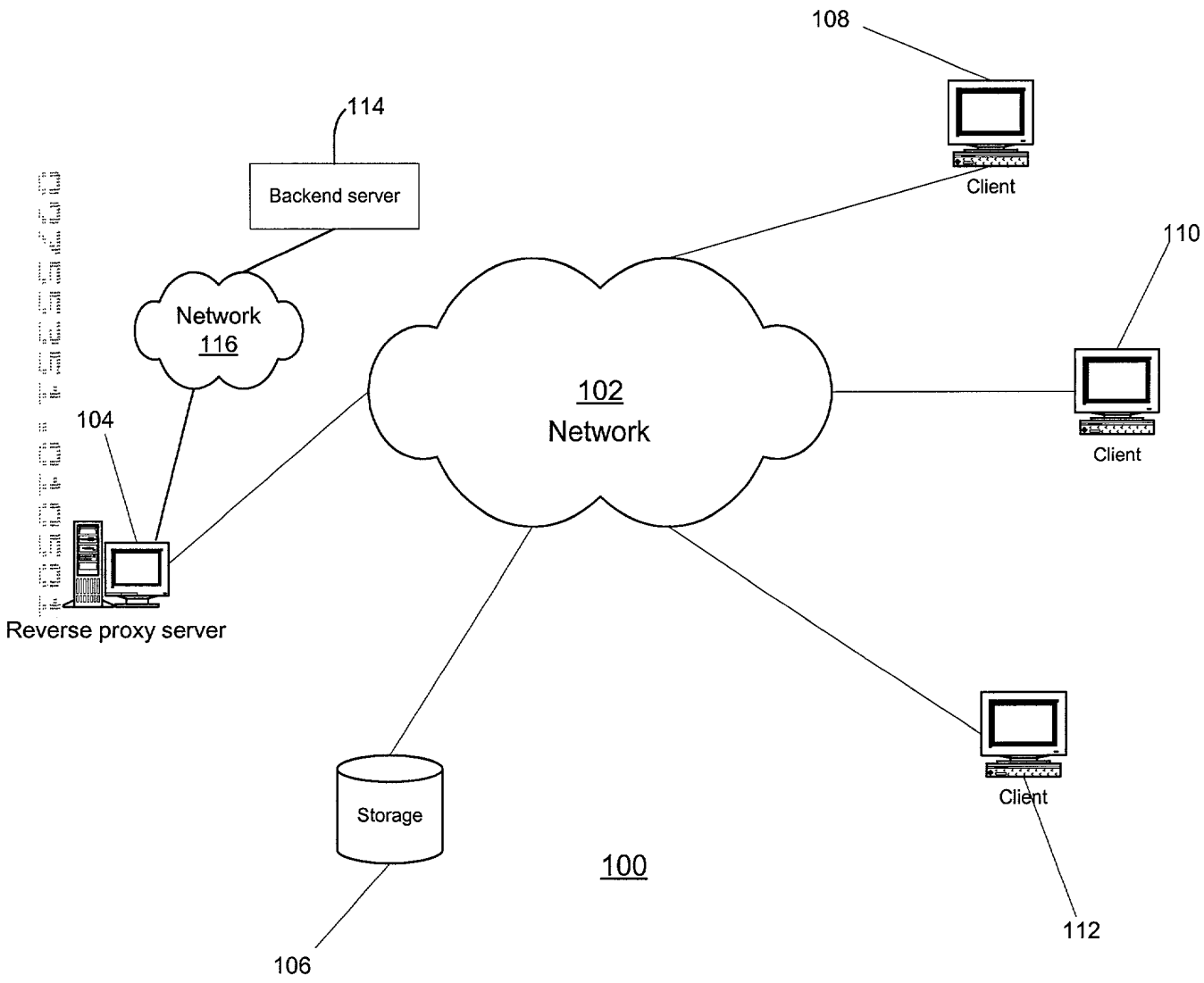
Figure 1


108
Client
114
Backend server
110
Network
116
102
Network
104
Client
Reverse proxy server
Storage
Client
100
106
112

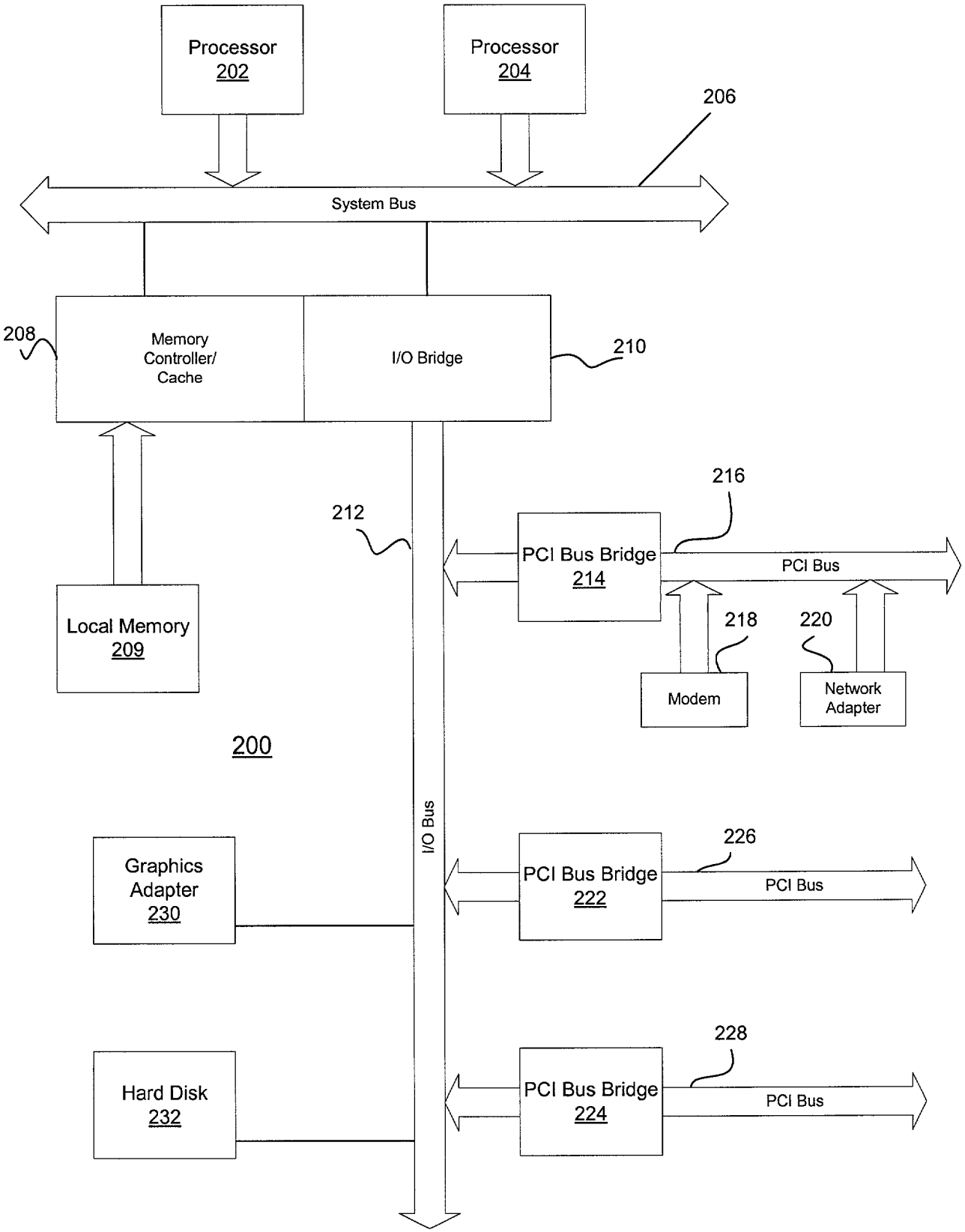

Processor
202
Processor
204
206
System Bus
208
Memory
Controller/
Cache
I/O Bridge
210
212
216
PCI Bus Bridge
214
PCI Bus
Local Memory
209
218
220
Modem
Network
Adapter
200
I/O Bus
Graphics
Adapter
230
226
PCI Bus Bridge
222
PCI Bus
Hard Disk
232
228
PCI Bus Bridge
224
PCI Bus

Figure 2

Figure 3

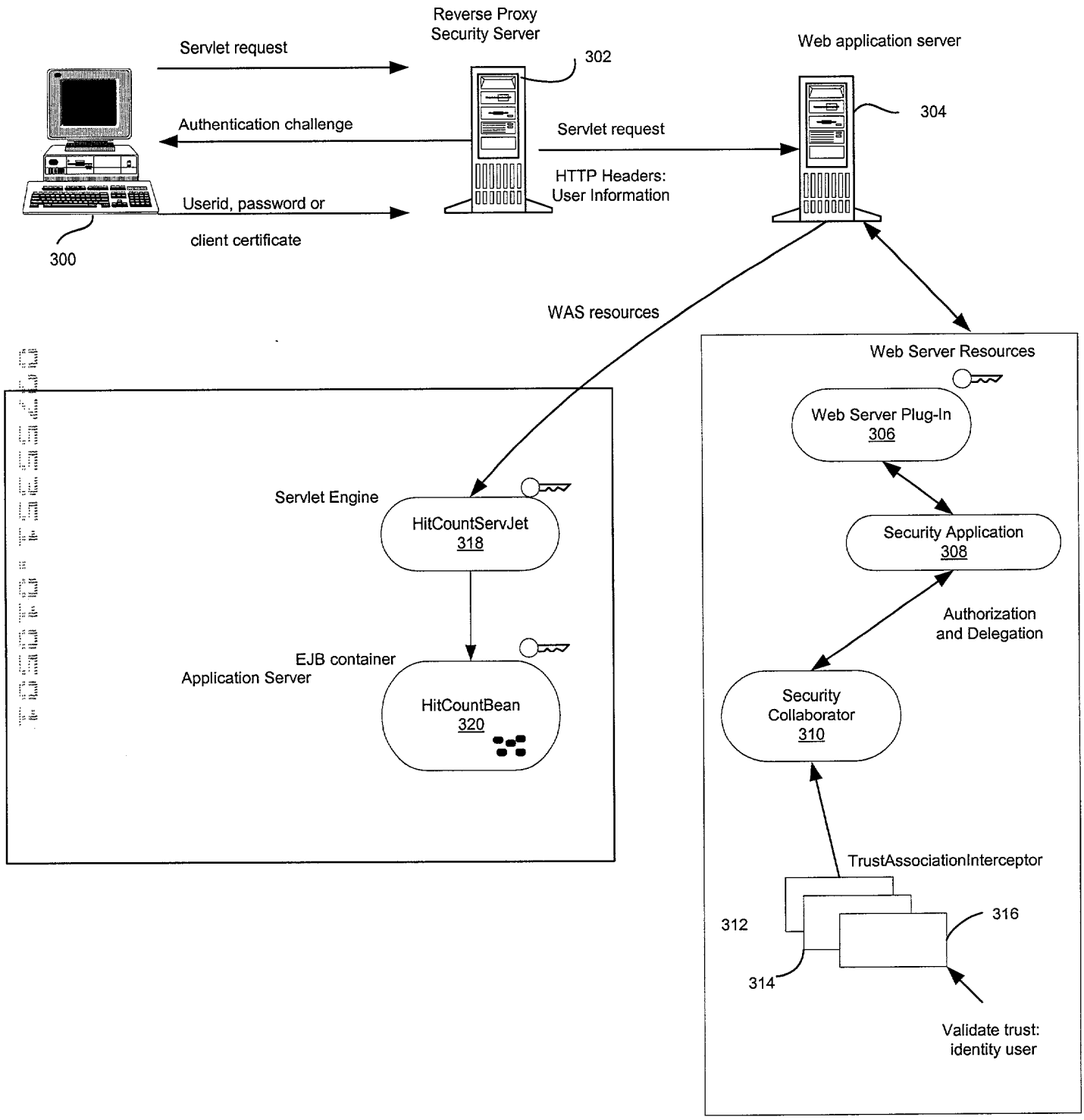

Reverse Proxy
Security Server
302
Web application server
304
Servlet request
Authentication challenge
Servlet request
HTTP Headers:
User Information
Userid, password or
client certificate
300
WAS resources
Web Server Resources
Web Server Plug-In
306
Servlet Engine
HitCountServJet
318
Security Application
308
Authorization
and Delegation
EJB container
Application Server
HitCountBean
320
Security
Collaborator
310
TrustAssociationInterceptor
312
316
314
Validate trust:
identity user

Figure 4A

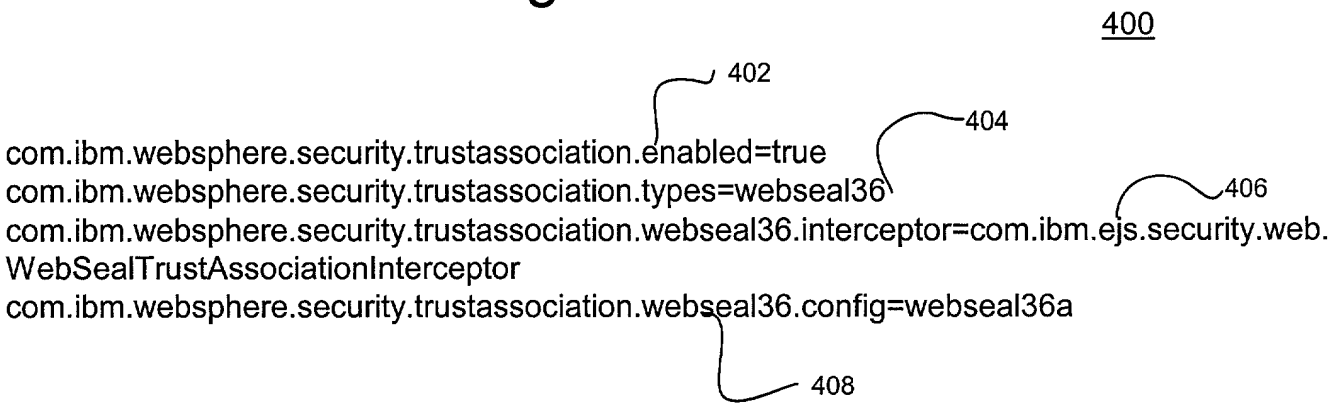


Figure 4B

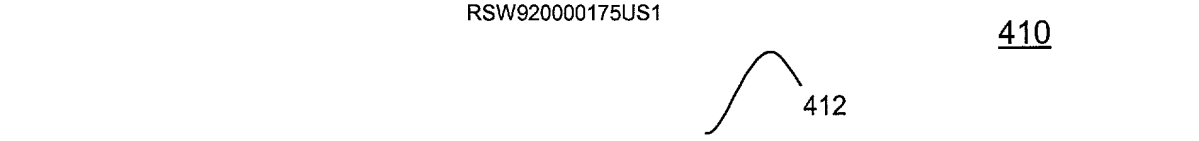
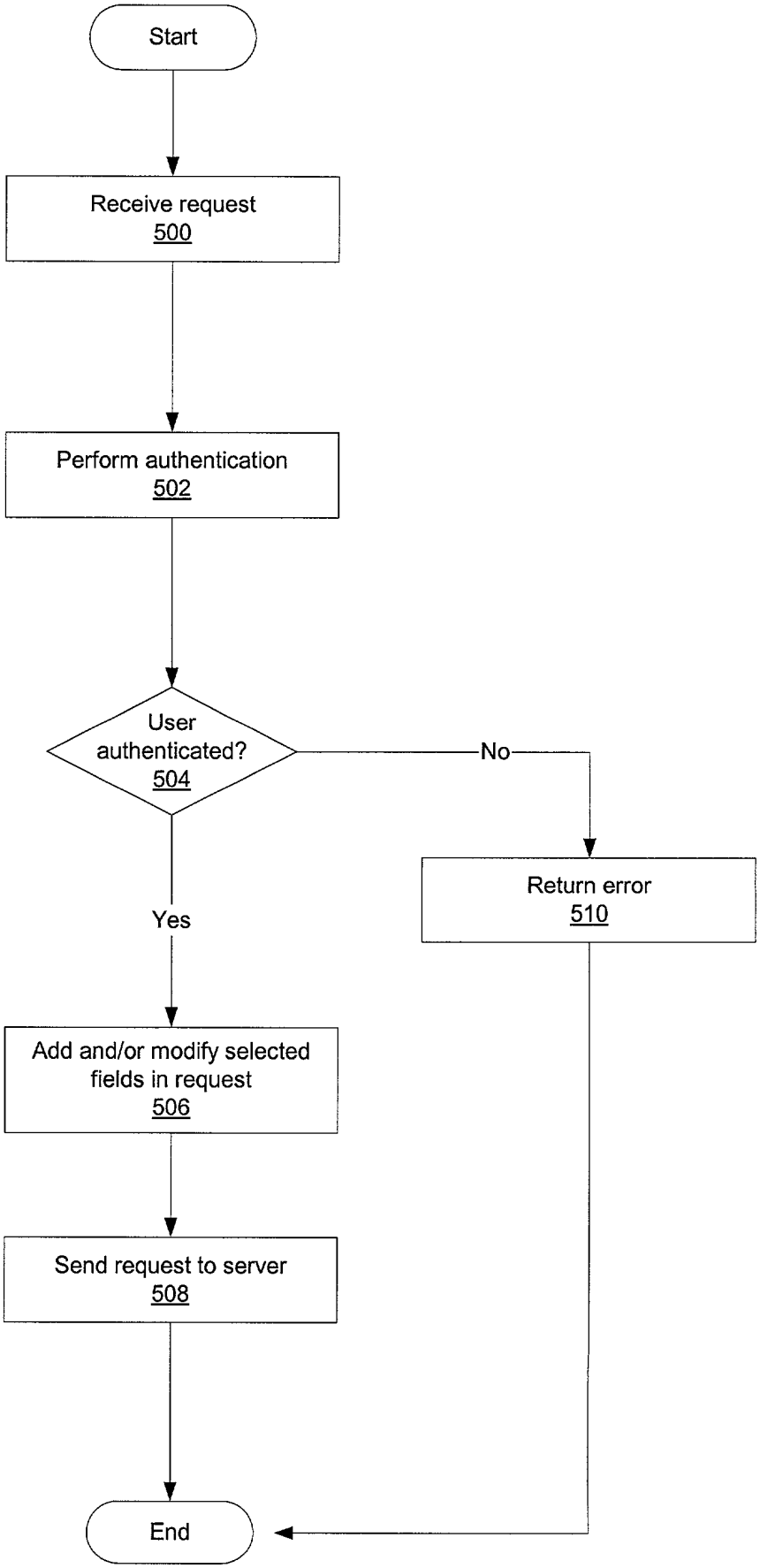


Figure 5


Start
Receive request
500
Perform authentication
502
User authenticated?
504
No
Return error
510
Yes
Add and/or modify selected fields in request
506
Send request to server
508
End

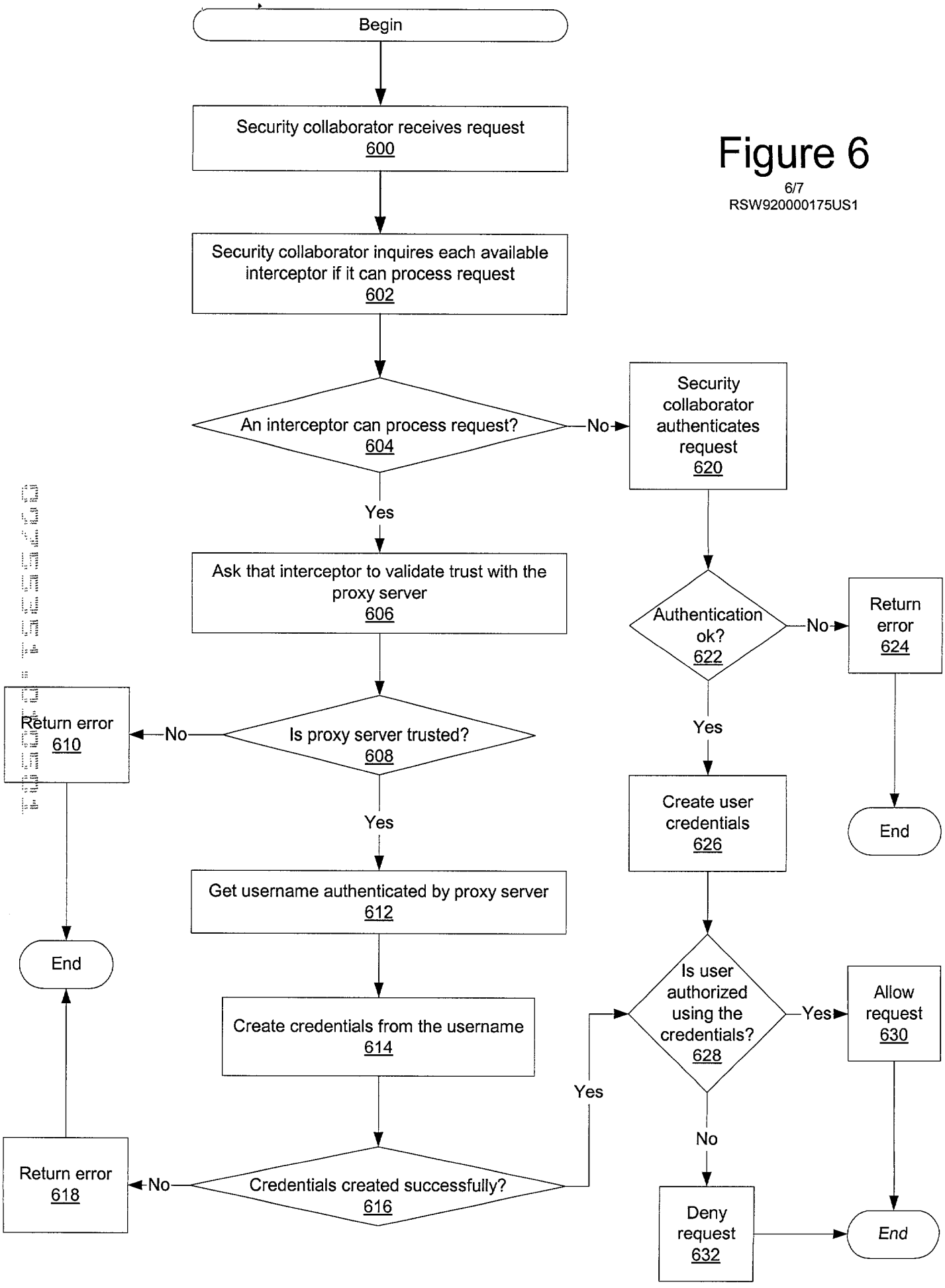

Begin
Security collaborator receives request
600
Security collaborator inquires each available interceptor if it can process request
602
An interceptor can process request?
604
No
Security collaborator authenticates request
620
Yes
Ask that interceptor to validate trust with the proxy server
606
Authentication ok?
622
No
Return error
624
Yes
Return error
610
No
Is proxy server trusted?
608
End
Create user credentials
626
Yes
Get username authenticated by proxy server
612
End
Is user authorized using the credentials?
628
Yes
Allow request
630
Create credentials from the username
614
Yes
No
Return error
618
No
Credentials created successfully?
616
Deny request
632
End

Figure 6

Figure 7

700

```
package com.ibm.websphere.security.web;
public interface TrustAssociationInterceptor
{
/**
Every interceptor should know which HTTP requests originate from
the third party server that it is supposed to work with.
Given an HTTP request, this method must be used to determine whether
or not this interceptor is designed to process the request, in
behalf of the trusted server it is designed to interoperate with.
If the return value is false or an exception is thrown, then
WebSphere will consider that the request is not routed via the
trusted proxy server the interceptor is designed to handle.
WebSphere will pass the request to the next interceptor till there
is no more interceptors available, in which case it will be treated
to be a directly submitted request.

**/
public boolean isTargetInterceptor(HttpServletRequest req)
throws WebTrustAssociationException;
/**
This method is used to determine whether the interceptor trusts the
server through which the request has been routed. This may involve
authenticating the server in some manner. All the required
Information to perform this operation should be available in the
HTTP request.
If the third party server failed the validation, or is unable to
provide the required information, a
WebTrustAssociationFailedException must be thrown. This would be
treated as an authentication failure and WebSphere would deny
access to the requested secure resource.
**/
public void validateEstablishedTrust (HttpServletRequest req)
throws WebTrustAssociationFailedException;
/**
This method is used to retrieve the username of the end client
(or the originator of the HTTP request). This method will be
invoked if the validateEstablishTrust method invocation was
successful.
The method returns a string. A return value of null or a
WebTrustAssociationUserException should be thrown if the username
is not available in the request header or the implementation
determines that the username provided was invalid (based on some
criteria, e.g., a list of valid usernames may have been
decided earlier).
**/
public String getAuthenticatedUsername (HttpServletRequest req)
throws WebTrustAssociationUserException;
```

702

704

706